**ECAI 2014 - International Conference – 6th Edition**
Electronics, Computers and Artificial Intelligence
23 October -25 October, 2014, Bucharest, ROMÂNIA

# A New Model for Interactions between Robots in a Swarm

Cătălin Buiu, Mihai Gânsari
Laboratory of Natural Computing and Robotics
Department of Automatic Control and Systems Engineering
Politehnica University of Bucharest
Bucharest, Romania
catalin.buiu@acse.pub.ro, mihai.gansari@acse.pub.ro

***Abstract* – Robotic swarms have a great potential to solve in a simple way, without the need for a central controller, a wide range of difficult real-world problems, such as transportation, search and rescue missions etc. While there are important results in what regards the emergence of complex behaviors from very simple interacting robots, the security of robotic swarms remains a problem which did not get the proper consideration. With the overall goal of enabling the full distributed security of a robotic swarm, the contributions of this paper are threefold. First, a bioinspired framework based on membrane computing (P colonies) is proposed for approaching this issue. Secondly, the functionalities of a P colonies simulator are presented and finally, the notion of P swarm is introduced as a relevant new formal model of safe interactions between robots in a swarm.**

***Keywords-multi-robot systems; swarms; security; membrane computing; P colonies***

## I. INTRODUCTION

The huge potential of multi-robot systems which operate on the principles of swarm intelligence, i.e. with a large number of (simple) robots interacting locally and with the environment, and without the need for a central controller to allocate tasks and give precise instructions, is confirmed by recent applications such as warehouse transportation and management, detection of gas or radioactive leaks, search and rescue, disaster relief, environmental monitoring, and military applications.

Over the last decade, increasingly sophisticated threats to our computers' and networks' security have emerged and are threatening our personas and digital identities, commercial secrets, credit card numbers, etc. Battling such complex and very often cryptic attacks is a difficult endeavor. In this context, there is an increasing concern on the security of human-robot and robot-robot relationships and physical interactions, but the security of robotic swarms has been largely overlooked.

The main goal of our research in swarm robotics is the development of a biologically inspired approach for enabling the distributed security of a robotic swarm, by building mechanisms that allow robots and swarms to possess or to develop an (innate) immunity towards a wide range of security hazards. This would allow a robotic swarm to display abilities such as constructing a model for malevolent behaviors, recognizing an intruder robot/swarm, expelling intruders, and correcting malicious behaviors by integrating intruders "on the right way" (in the original benevolent swarm).

The bioinspired formalism of membrane computing (P colonies more specifically) is proposed here as the foundation of an integrated approach for modeling security hazards and dealing in an efficient way with these threats to the swarm security. Other original contributions are the development of a P colonies simulator (the first reported in the literature) and the introduction of the concept of P swarm as a relevant model for interactions in a robotic swarm.

Section II gives a brief critical overview of swarm robotics in terms of benefits and security challenges. Section III presents the basics of P systems and P colonies, while Section IV offers the main contributions of this paper. Section V gives some conclusions and directions for further improvements.

## II. SWARM ROBOTICS. BENEFITS AND CHALLENGES

The idea of using teams or "swarms" of simple autonomous robots to collectively solve tasks gained momentum in the last decade. Robotic swarms have proved to be able to solve, for example, search and localization problems in laboratory settings, but very few applications have been reported in real-world environments. Consider for example the potential advantages a robotic swarm solution for real industrial problems, such as detection of gas or radioactive leaks, search and rescue, transportation of (dangerous) objects, etc. may offer. Such a system is fault-tolerant, has a low maintenance cost (because of using simple robots), is scalable (robots can be easily added), and flexible.

Sub-swarms are groups of robots in the swarm which try to achieve different goals. For example in Fig.1 sub-swarm 2 may search for a target, while sub-swarm 6 transports an object or searches for another target. Sub-swarms themselves are composed of one or more neighborhoods. Neighborhoods belonging to

978-1-4799-5479-7/14/$31.00 ©2014 IEEE

the same sub-swarm are smaller groups of robots which help each other in achieving the common goal of the sub-swarm.

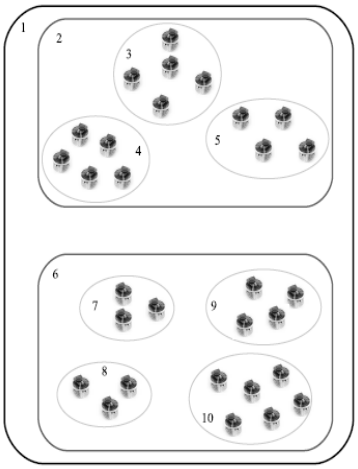


Figure 1. The hierarchical structure of a robotic swarm with 2 sub-swarms (2 and 6) and 7 neighborhoods (3, 4, 5, 7, 8, 9, 10).

For example, when searching for a target, the robots may group themselves in neighborhoods in order to help each other pass local minima and reach the target's location (neighborhoods 3, 4, and 5 in Fig. 1). Another example is the transportation of an object by groups of robots which are formed in order to carry and balance it (neighborhoods 7, 8, 9, and 10 in Fig. 1). Each robot belongs to one and only one neighborhood and therefore to one and only one sub-swarm. The swarm has a dynamic hierarchical structure due to the changes in the goal list, to the available robots and to other factors.

Particle Swarm Optimization (PSO) is a global stochastic optimization technique [1] and is used alone or in combination with other swarm intelligence algorithms such as ACO (Ant Colony Optimization) as a control algorithm for distributed robot swarms. Various mathematical models for swarm systems are reported in the literature [2]. In a robotic swarm, sub-swarms and neighborhoods may be used in order to model the robotic swarm hierarchy [3]. There are only a few approaches in the literature to evolve scalable control hierarchies, such as that in [4] which is using developmental evolution, specifically an L-system based approach. These results show that developmental evolution can be used to evolve hierarchical control structures that scale well but they do not go beyond the level of subswarm to the level of neighborhoods.

Human-robot interaction research is generally focused on the interaction of a human operator and a single complex robot, and so the methods developed for this approach cannot be applied for a swarm of robots. Furthermore, an operator cannot interact with each robot in the swarm. Human-swarm interaction has been largely overlooked and new techniques need to be developed to enable an intuitive and reliable swarm level interaction.

The security of a robotic swarm has been largely overlooked, too. As most of the robotic swarm implementations make use of wireless communications [5], what if malevolent users gain access to some of the robots in the swarm or what if foreign robots attach to the original swarm and possibly form neighborhoods or even sub-swarms which will eventually impose their own objectives (see e.g. Fig. 2 where a Khepera robot has malevolently joined a swarm of 10 e-puck robots)?

Other possible threats to swarm security are [6]: resource constraints (robots in a swarm are usually very simple entities), physical capture by adverse parties, lack of central monitoring and control, violation of explicit and implicit communication means, entity authentication, key management, and intrusion detection. Intrusion means the entry of an un-authorized entity (an adverse robot as in Fig. 2, a group of robots, or entire swarms of malevolent robots) which could manipulate the original goals of the robotic swarm. This situation asks for an immediate action (expelling of the un-authorized entities). As far as we are aware, there are no specific solutions to this problem and this requires an entirely new and integrated approach, from analyzing and modeling of security hazards to dealing with these attacks in a timely and efficient manner, so that the robotic swarm fulfills its original goals in a secure way.

Figure 2. An adverse robot has joined a swarm of 10 identical robots.

The reference papers for robotic swarm security are surprisingly only a few. Swarm engineering has been defined as the fusion of dependable systems (doing the right thing- achieving goals - and not doing the wrong thing - safety) engineering and swarm intelligence in [7] which considers a swarm robotics approach to physical containment or encapsulation. A related approach that explores fault-tolerance in robot swarms through Failure Mode and Effect Analysis and reliability modeling is presented in [8]. A first systematic attempt to categorize security challenges to swarm robotics is presented in [9].

The related field of secure authentication in robotics gained a special attention during the last years. A new field of study named artimetrics has been introduced in order to allow "identification, classification, and authentication of robots, software,

and virtual reality agents" [10]. However, the existing results build upon recent advances in biometric methods and so concentrate on humanoid robots, which are already looking very realistic and have a sum of human abilities (physical, cognitive, and emotional). More, theoretical aspects are somehow surpassed by practical applications. The problem of behaviors, misbehaviors, and security of a society of social robots is addressed in [11].

Involuntary and malicious behaviors and the possibility to detect them are considered by the authors, who also identify fundamental requirements for a system of behaviors for a society of robots: scalability, heterogeneity, reconfigurability, and security. They describe social behaviors as hybrid automata and consider only those behaviors for which models are a-priori available. Another paper [12] addresses the problem of multirobot systems security. However, the robots considered are humanoid ones showing considerable cognitive abilities, such as visual recognition. Cognitive robots (capable of inference, perception, and learning) and their behaviors are considered for building a reference model toward intelligent authentication [13].

## III. MEMBRANE COMPUTING

### A. P systems

Cell-like computing is one of the most promising paradigms of unconventional computation. Important theoretical results have been obtained during the last years, while practical applications still expect to gain momentum. P systems (PS) or membrane computing represent an active area of research in cell-like computing, drawing its inspiration from the membrane structure and functioning of a living cell. The basic idea of membrane systems [14] is to separate computing processes in different compartments (membranes) which are able to inter-communicate.

So, a P system (PS) represents a distributed and parallel computing model in which the basic data structures are multisets, strings or numerical variables. There are two main categories of PSs: hierarchical and tissue PSs. A hierarchical PS consists of several membranes placed inside a unique skin membrane, see Fig. 3. The analogy with the proposed hierarchical structure of a robotic swarm (Fig. 1) is obvious, for example neighborhoods being analogous to elementary membranes (that is, membranes not containing other membranes).

Regions have objects and rules assigned. The basic data structure in a typical PS is the multiset. Objects evolve by means of evolution rules which are localized, and are applied in a maximally parallel and non-deterministic manner. Most classes of PSs are equivalent to Turing machines, and so computationally complete/universal. Some practical applications of PSs were also proposed [15]. PSs are naturally parallel and distributed systems and have, therefore, a great potential for applications, especially in robotics. For example, membranes of a PS can be distributed over a grid or over a network of microcontrollers in a robot.

The computation done in each membrane region (i.e. the execution of membrane's rules) can also be done in parallel. In this sense, PSs can be used as a modeling paradigm for parallel and distributed control systems [16, 17] and is more suitable for hybridization with a PSO algorithm rather than other approaches, such as using fuzzy logic [18].

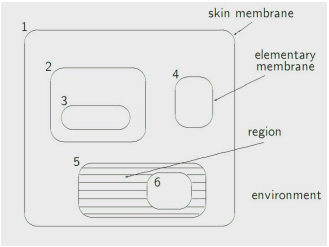


Figure 3. A P system's internal structure of membrane.

### B. P colonies

A P colony of capacity $c$ is a construct:

$$\Pi = (A, e, f, B_1, \ldots, B_n) \quad (1)$$

where $A$ is an alphabet (a set of objects), $e \in A$ is the basic object of the colony, $f \in A$ is the final object of the colony, $B_i \left(i = \overline{1,n}\right)$ are agents. Each $B_i$ is of the form $(O_i, P_i)$, where $O_i$ is a multiset of $c$ copies of the basic object $e$ (the initial state of the agent), and $P_i = \{p_{i,1}, p_{i,2}, \ldots, p_{i,k_i}\}$ is a finite set of programs. More, each program $p_{i,j}$ consists of $c$ rules. Usually, there are two types of rules: evolution rules of the form $a \to b$ (rewriting object $a$ into object $b$) and communication rules of the form $c \leftrightarrow d$ (object $c$ from the agent will move into the environment and object $d$ from the environment will move into the agent).

At the start of the computation performed by $\Pi$, the special basic objects exist only in the environment and in each agent. When no agent of $\Pi$ is able to fire any of its programs, the computation ends and the final result is the number of final objects in the environment.

A simple example of a P colony with one agent and two programs ($P_1$ and $P_2$) which executes the addition of number 1("+1") is presented in [19] where possible computation modes (parallel and sequential) for a P colony are also discussed. In formal terms, this simple P colony with one agent can be defined as:

$$AG_+ = (\{e, e\}, < e \to f; e \leftrightarrow l_+ >, < l_+ \to e; f \leftrightarrow e >) \quad (2)$$

Initially there are $n$ occurrences of the final object $f$ in the environment (Fig. 4) and after the execution of the two programs; there will be $n+1$ $f$ symbols in the environment.

A program where the agent can check for the presence in the agent of a given object $c$ is a checking

program [20] and it has the following form: $< a \to b; c \leftrightarrow d / c' \leftrightarrow d' >$.

After the evolution rule $a \to b$ is applied, if $c$ is present in the agent, then $c$ is exchanged with $d$ from the environment; if not, the exchange $c' \leftrightarrow d'$ will be performed.

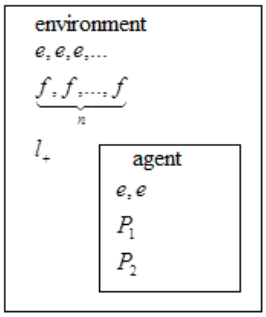


Figure 4. A simple P colony with one agent.

P colonies can compute whatever is algorithmically computable [20]. Further interesting results on the computing power are presented in the literature, together with a number of variants of P colonies. For example, homogeneous P colonies are P colonies with programs having the same type of rule for all objects inside an agent [21].

For P colonies of capacity two (i.e. two objects in each agent) insertion-deletion rules may be defined [22]: deletion programs of the form $< a, in; bc \to d >$ meaning that if $bc$ is inside the agent and $a$ is present in the environment, then the objects inside $bc$ are changed to $d$ and $d$ is brought into the agent; and insertion programs of the form $< a, out; b \to cd >$ where if $ab$ is inside the cell, then $a$ is sent out and $b$ is changed to $cd$. A sender is an agent with only insertion programs and a consumer is a cell with only deletion programs.

While in their first variants, P colonies had a passive environment, Eco-P colonies are extensions of P colonies with dynamical evolution of environments [23]. P colony automata are combining properties of finite automata and P colonies [15, 24]. An application of P Colonies automata for robot control is presented in [25].

## IV. P COLONIES AND ROBOTIC SWARMS

### A. Foundational idea

In order to address the issue of how to assure full physical and cyber-security of a robotic swarm through a systematic bio-inspired approach, two main objectives can be identified:

A. To rigorously analyze, define, classify and model the entire spectrum of robotic swarms' security hazards.

B. To propose, develop and test novel approaches for responding in a timely and appropriate manner to main sources of insecurity identified under (A).

The main idea on which all the research presented here is based is the natural similarity of membrane systems with robotic swarms (see also Fig. 1 and Fig. 3):

1. A P system is a system of biomembranes (a robotic swarm is a system of many simple robots grouped into neighborhoods and sub-swarms) which divide the internal cell space (the swarm as a whole) into discrete compartments (sub-swarms and neighborhoods) to segregate processes (tasks and sub-tasks) and components (robots), are selectively permeable (secure), and play a key role in organizing complex reaction sequences, energy conservation (power saving) and cell-to-cell (swarm-swarm or swarm-human) communication.

2. P colonies are based on the same ideas of swarm intelligence (to use simple agents, placed in a common environment, leading to non-trivial emergent behaviors).

3. P automata are combining features of classical automata and natural systems with a distributed architecture (like swarms).

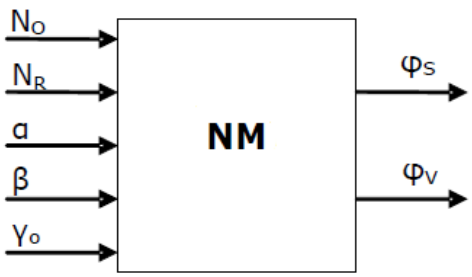


Figure 5. A neighborhood management module.

As an example, hierarchical numerical P systems [16] may be used for subswarms and neighborhoods management. For example, a neighborhood management (NM in Fig. 5) module may receive as inputs the number of objectives ($N_O$), the total number of robots ($N_R$), the minimum number of robots in a neighborhood (α), the minimum number of robots in a sub-swarm (β), and a vector containing the number of robots assigned to each objective ($\gamma_O$), and generates as outputs sets of addresses for each robot: $\varphi_S$, a vector describing the membership of a robot to a sub-swarm, and $\varphi_V$, a vector describing the membership of a robot to a neighborhood, dim $\varphi_V = N_R / \alpha$. Dynamic allocation of addresses may be obtained by using a numerical P system.

### B. A simulator of P colonies

The first practical step was the development of a P colonies simulator, the first of its kind. The simulator is written in Python and for the moment allows one to simulate a basic P colony with rewriting and exchange rules. In Fig. 6 a screenshot of the simulator running is presented.

The simulator will be extended with other rule formats and with the option of simulating Eco-P colonies and P automata.

### C. P Swarms

The notion of a P colony is extended here in order to apply it to robotic swarms.

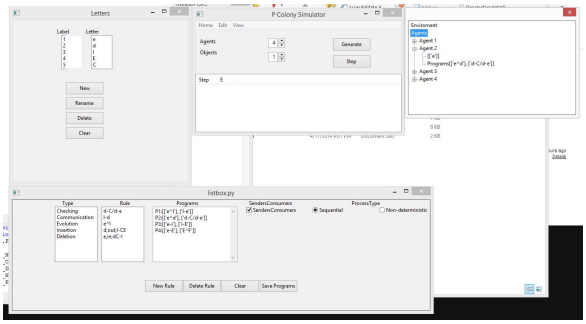

Figure 6. A simulator of P colonies.

A P swarm is a colony of P colonies in order to allow for each member P colony to model a subswarm of the original swarm. In order to assure the security of the swarm and to expel intruders (to not recognize them as partners in changing information in a direct peer-to-peer way or indirectly through the environment - that means "expelling" from a logical point of view) subswarms will be formed and evolved dynamically. For example, if there are $N$ robots in the swarm at the beginning, there will be $N$ subswarms (of one robot each) initially. Then based on some "adoption" rules, each robot will try to adopt robots in its subswarm. Those robots which will fail to do that will get adopted in other robots' subswarms. Consequently, there will be "abandoning" rules by which robots will not be members of the subswarm anymore. Finally, robots that will not get adopted by any of the P colonies will be considered adverse robots and any direct or indirect communication with these robots will cease.

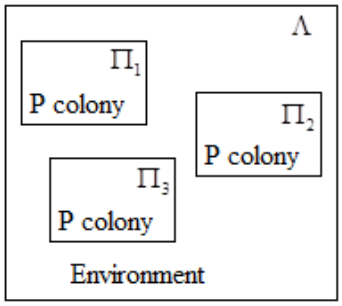


Fig. 7a. A simple P swarm ( $\Lambda$ ) with three P colonies inside ( $\Pi_1, \Pi_2, \Pi_3$ ).

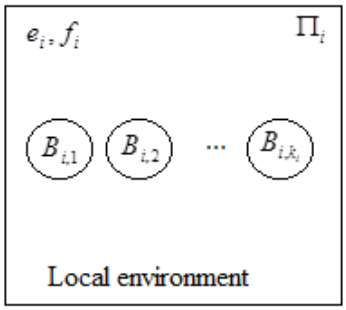


Fig. 7b. The structure of a member P colony ( $\Pi_i$ ).

Formally speaking, a P swarm is a construct of the form:

$$\Lambda = (A, e, f, \Pi_i) \quad (3)$$

where $\Pi_i (i = \overline{1, N})$ are P colonies of the form (1). As a simple example, Fig. 7a presents a P swarm with 3 P colonies inside (each one corresponding to a subswarm) and Fig 7b presents the structure of a member P colony, see also (1).

Rules inside each member P colony are of the following form in order to assure the proposed functionalities:

- Adoption rules: $a \to ab$ (robot $a$ will adopt robot $b$ in its subswarm);
- Abandoning rules: $cd \to c$ (robot $c$ will expel robot $d$ from its subswarm).

## V. Conclusions and Further Developments

This paper presented a work in progress whose main idea is to use P colonies (a class of P systems, or membrane computing) to model security issues for robotic swarms. A simulator of P colonies has been developed and a new model for interactions between robots in a swarm has been proposed, the so-called P swarm. Some directions for further research are as follows: to further extend and test the formal model based on P swarms, to propose new mathematical measures or criteria for evaluating the swarm behavior and different approaches to the control of robot swarms; to propose new variants of PSO-like algorithms that are amenable to hybridization with P systems; to further look how P colonies may be used in order to control a robotic swarm; to involve in the (complex) P-systems construction techniques from evolutionary programming (building membrane structures, evolution rules or entire P systems in an adaptive way); to check the correctness of P-systems based models (produced by evolution or programmed in a classic way) by means of model checking tools.

## Acknowledgment

This work was supported by a grant of the Ministry of National Education, CNCS-UEFISCDI, project number PN-II-ID-PCE-2012-4-0239. The work of Emilia Buhaev and Valentin Coteanu is acknowledged.

## References

[1] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proc. IEEE Int'l. Conf. on Neural Networks, IV, 1942–1948. Piscataway, NJ: IEEE Service Center, 1995.

[2] B. Varghese and G. McKee, "A mathematical model, implementation and study of a swarm system," Robotics and Autonomous Systems, Volume 58, Issue 3, 31 March 2010, pp. 287-294, doi:10.1016/j.robot.2009.08.006, 2010.

[3] C. Vasile, A.B. Pavel, and C. Buiu, "Integrating human swarm interaction in a distributed robotic control system," Proceedings of the 7th IEEE Conference on Automation Science and Engineering (IEEE CASE 2011), Trieste, 24-27 August, doi: 10.1109/CASE.2011.6042493, ISSN 2161-8070, 2011.

[4] T. Soule and R.B. Heckendorn, "A developmental algorithm for multi-agent swarms with scalable hierarchies," Proceedings of the 12 Annual Conference on Genetic and

evolutionary computation, GECCO-2010, doi: 10.1145/1830483.1830602, 2010.

[5] R L. Wenfeng and S. Weiming, "Swarm behavior control of mobile multi-robots with wireless sensor networks," Journal of Network and Computer Applications, Available online 21 March 2011, doi:10.1016/j.jnca.2011.03.023, 2011.

[6] Fiona Higgins, Allan Tomlinson, and Keith M. Martin, "Threats to the swarm: security considerations for swarm robotics," International Journal on Advances in Security, vol. 2, no. 2-3, pp. 288-297, 2009.

[7] Alan F.T. Winfield, Christopher J. Harper, and Julien Nembrini, "Towards dependable swarms and a new discipline of swarm engineering," in E. Sahin and W.M. Spears (Eds.): Swarm Robotic WS 2004, LNCS 3342, pp. 126–142, 2005.

[8] A. F. T. Winfield and J. Nembrini, "Safety in numbers: fault–tolerance in robot swarms," International Journal of Modelling, Identification and Control, vol. 1, no. 1, pp. 30–37, 2006.

[9] F. Higgins, A. Tomlinson, and K. Martin, "Survey on security challenges for swarm robotics," in Fifth International Conference on Autonomic and Autonomous Systems (ICAS 2009), R. Calinescu, F. Liberal, M. Marin, L. P. Herrero, C. Turro, and M. Popescu, Eds. Valencia: IEEE Computer Society, April 2009, pp. 307– 312, 2009.

[10] Marina L. Gavrilova and Roman V. Yampolskiy, "State-of-the-art in robot authentication," IEEE Robotics and Automation Magazine, 17(4):23 - 24, December 2010.

[11] A. Bicchi, A. Fagiolini, and L. Pallottino, "Towards a Society of Robots: Behaviors, Misbehaviors, and Security," IEEE Robotics and Automation Magazine, 17(4):26 - 36, December 2010.

[12] Javier Ruiz-del-Solar, Rodrigo Verschae, Matias Arenas, and Patricio Loncomilla, "Play Ball! Fast and accurate multiclass visual detection of robots and its application to behavior recognition," IEEE Robotics and Automation Magazine, vol. 17, no. 4, pp. 43 -53, 2010.

[13] Y. Wang, "Cognitive robots," IEEE Robotics and Automation Magazine, vol. 17, no. 4, pp. 54-62, 2010.

[14] Gh. Paun, "Computing with membranes," Journal of Computer and System Sciences, vol. 61, pp. 108–143, Elsevier, 2000.

[15] Gh. Paun, G. Rozenberg, and A. Salomaa, The Oxford Handbook of Membrane Computing, Oxford University Press, 2010.

[16] C. Buiu, C. Vasile, and O. Arsene, "Development of membrane controllers for mobile robots," Information Sciences, Vol. 187, 15 March 2012, pp. 33-51, doi:10.1016/j.ins.2011.10.007, ISSN 0020-0255, 2012.

[17] Ana Pavel and C. Buiu, "Using enzymatic numerical P systems for modeling mobile robot controllers," Natural Computing, Vol. 11, Issue 3, pp. 387-393, 2012.

[18] G.K. Venayagamoorthy, L.L. Grant, L.L. and S. Doctor, "Collective robotic search using hybrid techniques: Fuzzy logic and swarm intelligence inspired by nature," Engineering Applications of Artificial Intelligence, Volume 22, Issue 3, pp. 431-441, 2009.

[19] J. Kelemen, A. Kelemenová, "On P colonies, a biochemically inspired model of computation," in Proc. of the 6th International Symposium of Hungarian Researchers on Computational Intelligence, Budapest TECH, Hungary, pp. 40-56, 2005.

[20] J. Kelemen, A. Kelemenová, G. Păun, "Preview of P colonies: A biochemically inspired computing model," in M. Bedau et al. (eds.) Workshop and Tutorial Proceedings, Ninth International Conference on the Simulation and Synthesis of Living Systems, ALIFE IX, Boston, Mass, pp. 82-86, 2004.

[21] L. Cienciala, L. Ciencialová, A. Kelemenová, "Homogeneous P colonies," Computing and informatics 27, pp 481-496, 2008.

[22] L. Ciencialová, E. Csuhaj-Varjú, A. Kelemenová, G. Vaszil, "On very simple P colonies," in Proceeding of The Seventh Brainstorming Week on Membrane Computing, Sevilla, Spain, vol. I, pp. 97-108, 2009.

[23] L. Cienciala, L. Ciencialová, "Eco-P colonies," in G. Păun, M.J. Pérez-Jiménez, A. Riscos-Núñez, G. Rozenberg, A. Salomaa, Eds. WMC 2009, LNCS, vol. 5957, pp. 201-209, Springer, Heidelberg, 2010.

[24] L. Cienciala, L. Ciencialová, E.C. Varjú, G. Vaszil, "PCol automata: recognizing strings with P colonies," in Eighth brainstorming week on membrane computing, Sevilla, 2010. (RGNC Report 01/2010)

[25] M. Langer, L. Cienciala, L. Ciencialová, M. Perdek, A. Kelemenová, "An application of the PCol automata in robot control," in 11th Brainstorming Week on Membrane Computing, pp. 153–164, 2013.